

VENTURUS 4TECH



BONUS

VENTURUS4TECH

ES6



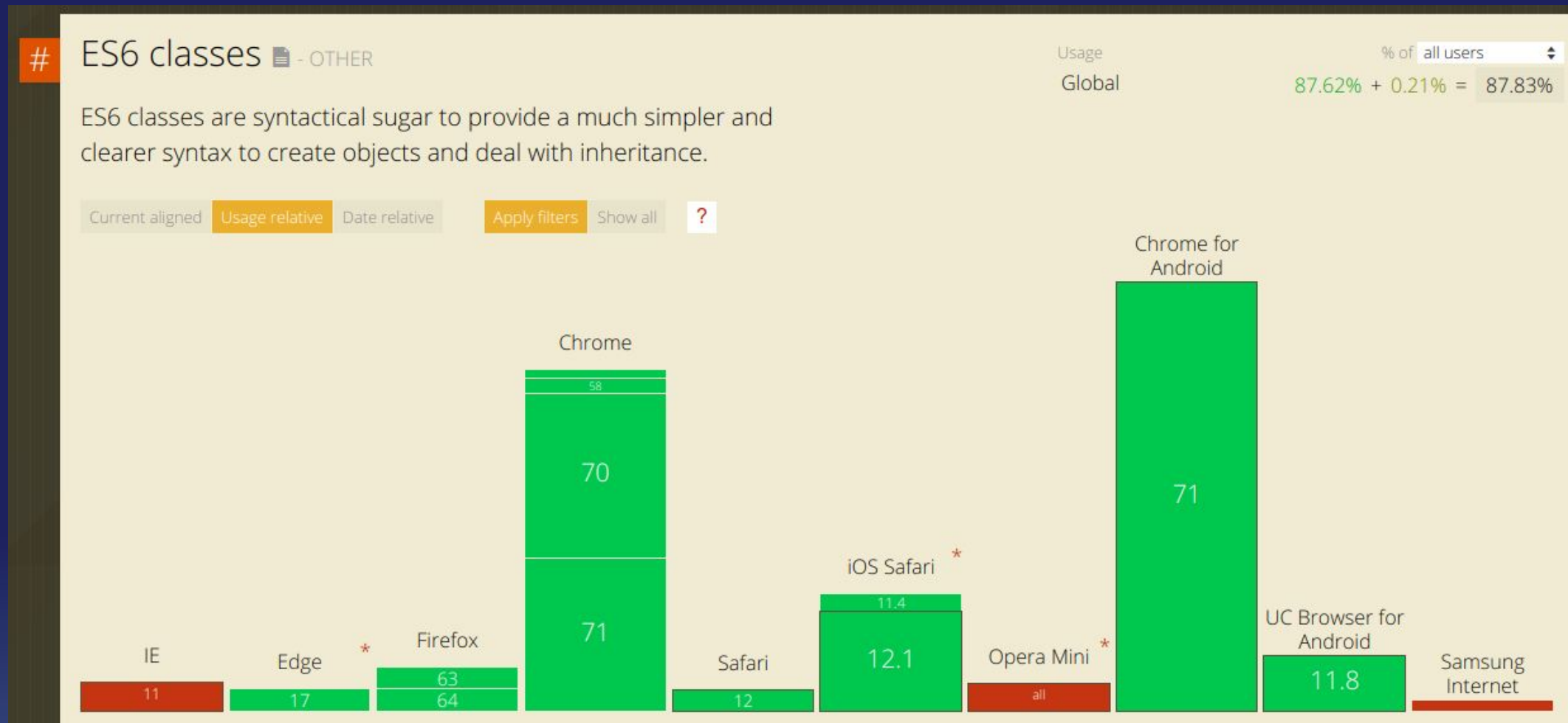
ES6

- EcmaScript é o nome oficial do JavaScript
- ECMAScript 6 também é conhecido como ES6 e ECMAScript 2015
- Há um comitê (TC39) responsável pela manutenção e evolução das especificações
- Releases anuais
- Atualmente na versão ES9 (EcmaScript 2019 - junho/2018)



Sobre ES6: [Site oficial](#)

SUPORTE - Can I Use?



Site CanIUse: [Suporte ao ES6 pelos navegadores](#)



ME

JAVASCRIPT

**Strongly Typed
inherently Safe
Compiled
Programming
Languages**

Variáveis

- **const**
 - valor não pode ser modificado
- **let**
 - variável que existe apenas no bloco, no contexto em que foi criada
- **var**
 - variável global, serve para a aplicação ou classe



Saiba mais: [Blog do Alura - Entenda a diferença entre var, let e const no JavaScript](#)

Arrow functions

- Não cria um novo contexto, pode acessar o this
 - (`<nomeParametro>`) `=>` { `<lógica>` }
 - `function(<nomeParametro>) { lógica }`



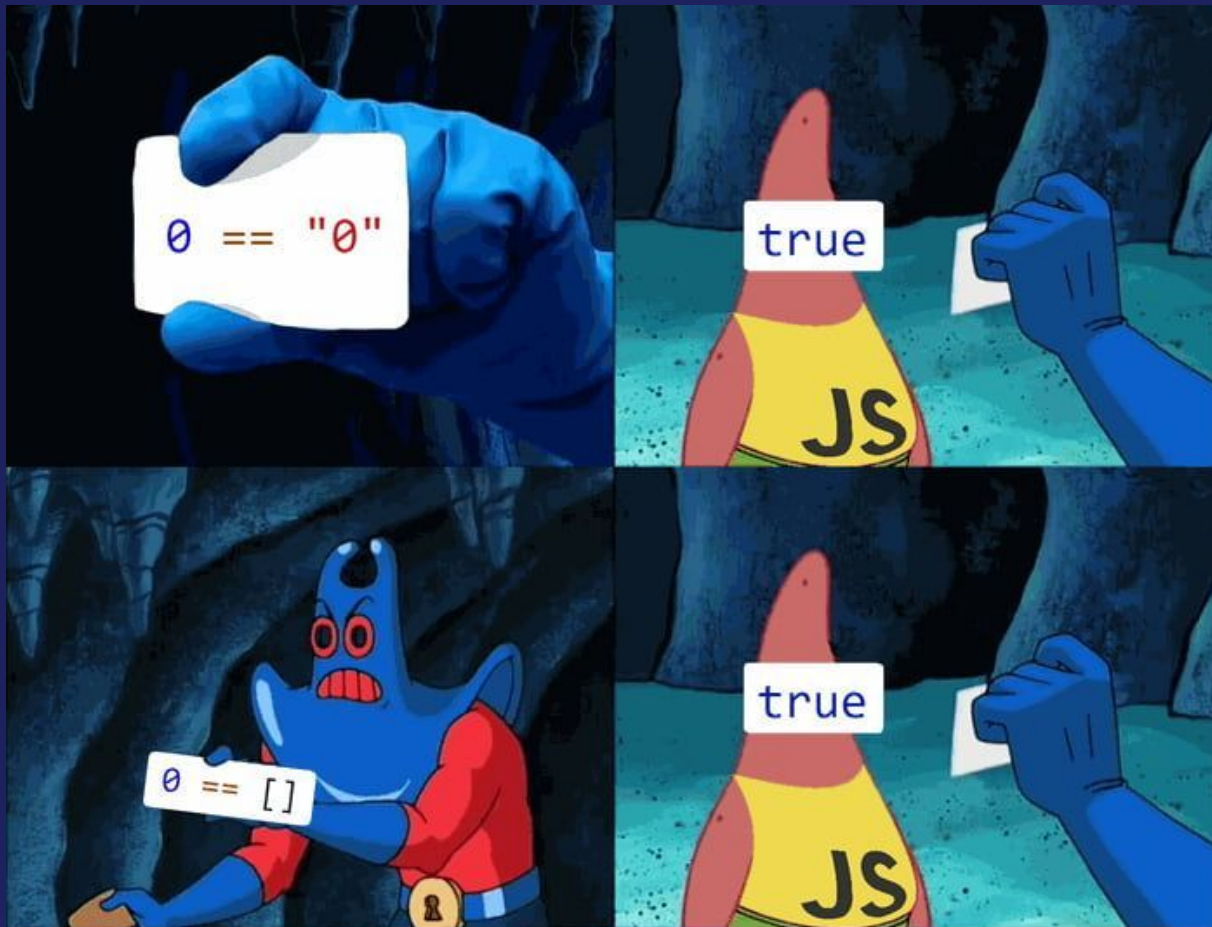
Sobre Arrow Functions: [ES6: arrow functions em 5 minutos](#)

+conceitos

- **Operador “=”**
 - atribui valor
 - `x = 1`
- **Operador “==”**
 - compara valores
 - `x == 1 // true`
 - `x == “1” // true`
- **Operador “===”**
 - compara valores e tipos
 - `x === 1 // true`
 - `x === “1” // false`



Sobre operadores (bem completo): [Expressões e operadores](#)



+conceitos

- **Array functions**

- Métodos de iteração

- `find(elemento => { elemento.sexo == 'M' })`
 - encontra o primeiro item que corresponde à condição
 - `filter(elemento => { elemento.sexo == 'M' })` : encontra todos os itens que correspondem à condição
 - `map((elemento, indice) => { elemento.index = indice })` : transforma os itens de acordo com a lógica
 - e muitos outros



Sobre funções de array (bem completo): [Array functions](#)

+conceitos

- **Referenciar atributos**
 - objeto.atributo = '123'
 - objeto['atributo'] = '123'
- **Console (debugging)**
 - console.log()
 - console.error()
 - console.info()
 - console.trace()



Sobre console: [10 Tips for Javascript Debugging Like a PRO with Console](#)

+conceitos

- **Objetos e arrays são sempre copiados com referência**

- `a = { item: 1 }; // a.item == 1 e b == undefined`
- `b = a; // a.item == 1 e b.item == 1`
- `b.item = 2; // a.item == 2 e b.item == 2`

Usar `Object.assign()` para copiar valor

Usar spread operator para copiar valor

- **Spread operator**

- “Abre” o objeto com os valores
- `b = { ...a }`
- `b = { item: 1 }`



Saiba mais: [Object.assign\(\)](#), [Spread operator](#)



JAVASCRIPT

JAVASCRIPT EVERYWHERE

makeameme.org



Mãos na massa

- Vamos testar alguns desses conceitos agora no console do ChromeDevTools





+Links e referências

- Para estudar mais:
 - [16 JavaScript Concepts JavaScript Professionals Must Know Well](#)
 - [The Modern JavaScript Tutorial](#)
 - [10 JavaScript concepts you need to know for interviews](#)



**Missão #DIA5
concluída!**



